3D Camera Trajectory Based on multiple ArUco Markers

Krutikaben Nareshbhai Patel
Department of Computer Science
University of Montreal
krutikaben.nareshbhai.patel@umontreal.ca

Abstract

We introduce the problem of 3D camera trajectory, which involves predicting the trajectory of a moving object across a cameras. The main purpose of this project is finding the object's position and orientation, referred to as the pose. Vision-based localization(VBL) is a technique where a camera is used to detect landmarks in the environment and calculate a object's pose from that information. A common method within VBL is fiducial marker tracking. To estimate the object pose in environment, we will be using ArUco markers.

1 Introduction

A great amount of research has been conducted with the goal of developing alternative solutions to the positioning problem. Examples of vision-based positioning which utilizes one or more cameras. Vision-based localization (VBL) is an approach that often takes advantage of visual landmarks and computer vision to extract information. Fiducial markers are specifically constructed with the purpose of encoding/decoding information in the marker and tracking of the marker. A benefit of this approach is that the marker can be designed specifically to be easily detectable by a camera.

Vision-based tracking can be used in a variety of applications, which may be reliant on computational speed to achieve real-time video capturing, robustness to occlusion in cases of changing environments, or pose estimation accuracy. Due to the large amount of algorithms developed to perform vision-based tracking, it is crucial to evaluate the different variations. This is necessary in order to determine which is suitable for each application without spending excessive amounts of time browsing through blogs, literature and complex research papers.

Compared to markerless pose tracking, a fiducial marker called ArUco provides a fast and accurate solution to the problem. With these advantages, this marker-based technique is ready to be used in virtual reality and operate in low-cost wearable devices. Therefore, we will be using ArUco marker for object pose estimation and tracking known as 3D camera trajectory.

2 Algorithms

2.1 Fiducial Markers

Fiducial markers are designed to contain an ID while having a suitable structure for reliably and efficiently extracting information such as pose estimation. Most designs share the same working principles, however they each differ in various ways to achieve optimal performance for their designated purpose, as one application may prioritize processing time while another prioritizes a more accurate pose estimation.

2.1.1 ArUco Markers

The ArUco marker is a square binary fiducial marker that is commonly used in order to determine position. It is fast, which make it a good choice for video capturing, and the binary square markers are easily detectable. The marker used in this application is a 5X5 grid with 100 mm, as the one in figure.

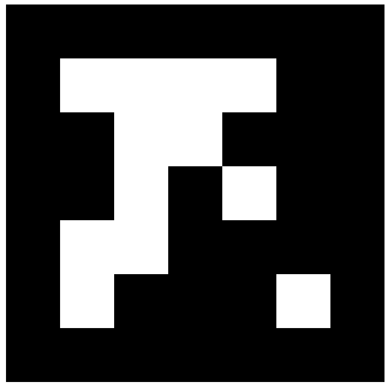

Figure 1: 5x5 ArUco Marker with ID=12

Marker Detection: The creators of the library describe the steps for marker detection, starting with a camera calibration step to obtain the camera's parameters which are needed to correctly determine where a 3D point in space projects to the camera.

Camera lenses normally capture the image with some distortion. Figure 2 demonstrates how the marker from figure 1 would look like with image distortion. This shows how a square marker may not appear to be square in the image frame. This needs to be taken into consideration and compensated for to improve marker detection.

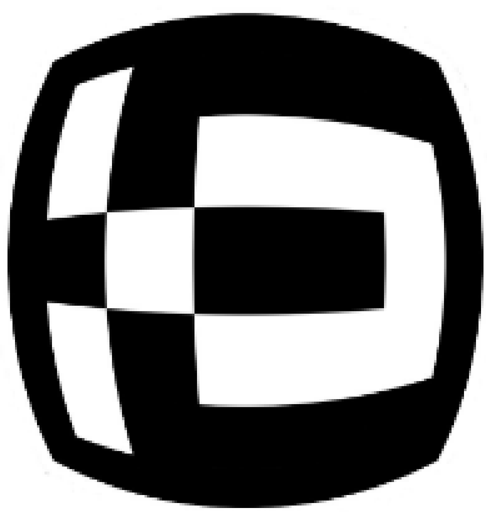

Figure 2: Distorted ArUco Marker

The marker detection starts with obtaining borders through adaptive thresholding. This means that only features satisfying certain threshold conditions are captured, and the threshold is adaptive due to possible variations in illumination for example. The borders are used to find contours. Borders that do not fulfill certain requirements that make them a possible match for a marker are filtered out. Remaining shapes are compared, and if two rectangles are too close to each other, one is removed.

This is primarily used to remove the inner rectangle that is detected on the markers, leaving only an outer border surrounding the marker content.

The first step of the marker identification process is to perform a homography transformation to remove the projection perspective and get the frontal view of the marker.

2.2 Pose Estimation

Pose estimation is of great importance in many computer vision applications: robot navigation, augmented reality, and many more. This process is based on finding correspondences between points in the real environment and their 2d image projection. This is usually a difficult step, and thus it is common to use synthetic or fiducial markers to make it easier.

2.3 ROS Transform

The main goal of this algorithm is to Maintain relationship between multiple coordinate frames overtime and Transform points, or vectors between two coordinates.

A main two main tasks that users generally use tf for transform between coordinates: broadcasting and listening.

2.3.1 Broadcasting transforms:

It publish the relative pose and coordinate to the system. Which allow us to setup our own relationship between two coordinate frames

2.3.2 Listening transforms:

It specify the published transform and query the specific transform between coordinate frames (not quite the same as Subscribing to a Topic).

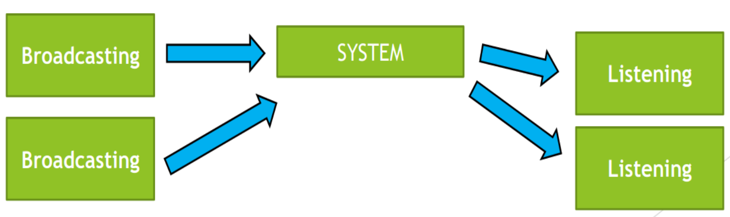

Figure 3: ROS Transform

3 Discussion

For 3D camera trajectory pose estimation is very important, because until the pose is not estimated correctly we won't be able to broadcast the TF. Another major part of this project is to perform the camera calibration. After detecting the ArUco marker, we calibrate the camera to get the camera parameters. We can estimate the object pose only after the camera calibration.

Why Camera calibration is Important?
A camera's job is to convert light that hits its image sensor into an image. An image is made up of picture elements known as pixels. In a perfect world, these pixels (i.e. what the camera sees) would look exactly like what you see in the real world. However, no camera is perfect. They all contain physical flaws that can, in some cases, cause significant distortion.

For example, Imagine we have a robotic arm that is working inside a warehouse. Its job is to pick up items off a conveyor belt and place those items into a bin. To help the robot determine the exact position of the items, there is a camera overhead. To get an accurate translation between camera pixel coordinates and real world coordinates, we need to calibrate the camera to remove any distortion. By calibrating a camera, we can enable a robotic arm, for example, to have better "hand-eye" coordination. We can also enable a self-driving car to know the location of pedestrians, dogs, and other objects relative to the car.

In this project, we have used 10-15 chessboard images with different angles to get the accurate camera parameters to estimate the correct pose of the markers. After pose estimation, we have publish the tf between the camera and marker. To broadcast tf we have used the ROS 2 Galactic and Foxy. And to visualize the trajectory we have used the Rviz2.

4 Tools and Framework

4.1 Pycharm Professional

We have used the python as our core language, so we have used the pycharm professional IDE for project implementation.

4.2 OpenCV

OpenCV is an open source computer vision software library. The library contains more than 2500 algorithms that, among many other things, can identify objects, track camera movements and detects markers.

4.3 ArUco Marker

An ArUco marker is a synthetic square marker composed by a wide black border and an inner binary matrix which determines its identifier (id). The black border facilitates its fast detection in the image and the binary codification allows its identification and the application of error detection and correction techniques. The marker size determines the size of the internal matrix.

4.4 ROS 2 Foxy

The Robot Operating System (ROS) is a set of software libraries and tools for building robot applications. From drivers to state-of-the-art algorithms, and with powerful developer tools, ROS has what you need for your next robotics project. And it's all open source.

4.5 Rviz2

RVIZ is a ROS graphical interface that allows you to visualize a lot of information, using plugins for many kinds of available topics. RVIZ2 is simply a port of RVIZ into ROS2.

5 Conclusion

In this project, we have seen that it is reliable to perform 3D camera trajectory using ArUco marker, because it gives the more accurate pose estimation result. As a result, the broadcasting between camera and ArUCo marker is accurate.

References

ArUco Marker: *https://gitlab.com/labv3d/aruco*

ArUco Marker Generator: *https://chev.me/arucogen/*

Colcon Tutorial: *https://docs.ros.org/en/foxy/Tutorials/Colcon-Tutorial.html*

Detection of ArUco Markers: *https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html*

ROS 2 Documentation: *https://docs.ros.org/en/galactic/Installation/Ubuntu-Install-Binary.html*

ROS 2 Foxy Fitzroy: *https://github.com/ros2/ros2/releases*

ROS 2 ArUco Mapping: *http://wiki.ros.org/aruco_mapping*